

# Statistical Pruning for Near-Maximum Likelihood Decoding

Radhika Gowaikar, *Student Member, IEEE*, and Babak Hassibi

**Abstract**—In many communications problems, maximum-likelihood (ML) decoding reduces to finding the closest (skewed) lattice point in  $N$ -dimensions to a given point  $x \in \mathbb{C}^N$ . In its full generality, this problem is known to be NP-complete. Recently, the expected complexity of the sphere decoder, a particular algorithm that solves the ML problem exactly, has been computed. An asymptotic analysis of this complexity has also been done where it is shown that the required computations grow exponentially in  $N$  for any fixed SNR. At the same time, numerical computations of the expected complexity show that there are certain ranges of rates, SNRs and dimensions  $N$  for which the expected computation (counted as the number of scalar multiplications) involves no more than  $N^3$  computations. However, when the dimension of the problem grows too large, the required computations become prohibitively large, as expected from the asymptotic exponential complexity. In this paper, we propose an algorithm that, for large  $N$ , offers substantial computational savings over the sphere decoder, while maintaining performance arbitrarily close to ML. We statistically prune the search space to a subset that, with high probability, contains the optimal solution, thereby reducing the complexity of the search. Bounds on the error performance of the new method are proposed. The complexity of the new algorithm is analyzed through an upper bound. The asymptotic behavior of the upper bound for large  $N$  is also analyzed which shows that the upper bound is also exponential but much lower than the sphere decoder. Simulation results show that the algorithm is much more efficient than the original sphere decoder for smaller dimensions as well, and does not sacrifice much in terms of performance.

**Index Terms**—Maximum-likelihood decoding, multiple antenna systems, reduced complexity, sphere decoder.

## I. INTRODUCTION

MULTIPLE antenna communication systems have been shown to be capable of achieving high data rates. However, reliable decoding in these systems requires very high complexity. For a wide class of space-time transmission schemes (see, e.g., [1]–[3]) maximum-likelihood (ML) decoding requires us to solve an integer least-squares problem. This is the problem of finding the closest (skewed) lattice point in  $N$ -dimensions to a given point  $x \in \mathbb{C}^N$ , which is known in general to be NP-hard. Most existing communications systems employ approximations or heuristics and typically require  $O(N^3)$  operations (since underlying all the methods is the calculation of a pseudo-inverse). Zero forcing cancellation,

nulling, and canceling and nulling and canceling with optimal ordering [1], [2], [4] are some of these. However, the bit error rate (BER) performance of these is vastly inferior to that of the exact methods.

Exact methods that search over the entire (finite) signal-space require search over a space growing at an exponential rate. More sophisticated exact methods such as Kannan's algorithm [5], the KZ algorithm [6] and the sphere decoding algorithm of [7] attempt to reduce the search space. The branch and bound algorithm, popularly used to solve integer (usually linear) programming problems, could also be used [8]. However, branch and bound imposes additional constraints on the optimizing variables to reduce the size of the problem and also requires one to estimate upper and lower bounds for the objective function to prune the search tree. In [9], an improved sphere decoder based on the branch and bound method is proposed.

In the sphere decoding algorithm, we first determine all lattice points lying in a hypersphere centered at  $x$  and then determine the point closest to  $x$ . The complexity of the algorithm is, therefore, determined by the amount of work that is required to determine all lattice points inside a given hypersphere (for some alternatives to sphere decoding, see [6], [10], and [11]). It can be shown that, both from a worst-case and from an average point of view, the sphere decoding algorithm requires exponential complexity (see, e.g., [12]). In [13], an alternative viewpoint has been taken up where, since in communications problems the noise vector and the lattice-generating matrix are random, the computational complexity is viewed as a random variable. Analyzing the expected complexity of sphere decoding, as well as its second-order moment [13] shows that, over a wide-range of rates, dimensions, and SNRs, the algorithm uses no more than  $N^3$  multiplications. While this is a very interesting result, for large enough  $N$  and low SNRs, the expected number of operations becomes prohibitively large. This fact is formalized in [14] where it is shown that, for any SNR, the sphere-decoder has exponential expected complexity.

In spite of this, the sphere decoder has attracted great interest, and it has been proposed as the decoder for several space-time coded systems. In addition, several modifications to the sphere decoder have been suggested in the last few years that attempt to reduce the computation involved [15]–[21]. Implementations of the sphere decoder in a complex setting rather than a real one are suggested in [13] and [22]. Some of the suggested modifications solve the ML decoding problem exactly [16], [17], [19] and others sacrifice some performance in order to reduce complexity [15], [20], [21].

The efficiency of the sphere decoder with respect to other methods shows the power of the probabilistic viewpoint and we will continue to use it in this paper. The main point is to

Manuscript received August 26, 2005; revised July 10, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Franz Hlawatsch.

The authors are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 (e-mail: gowaikar@systems.caltech.edu; hassibi@systems.caltech.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2006.890912

understand the role of the randomness underlying the problem and leverage it suitably. Thus, we will propose a modification to the sphere decoding algorithm that uses statistical pruning to reduce the exponentially large search space to one that is much smaller yet contains the optimal solution with high probability. This causes a significant reduction in complexity, at the price of a slight increase in the BER. We present a bound on this loss of performance and describe how to control this loss. The complexity is analyzed in three different ways. The first analysis is for asymptotically large systems and is of theoretical interest. The other two are valid for any value of  $N$  and can be used to design and understand practical systems.

The remainder of the paper is organized as follows. In Section II, we introduce the integer least-squares problem and demonstrate that it arises in the ML decoding of multiple antenna systems. In Section III, the basic sphere decoding algorithm is explained and in Section IV the notion of complexity is outlined. We introduce the statistics of the problem and propose a new algorithm, *viz.*, the increasing radii algorithm, that exploits these statistics in Section V (this algorithm was first presented in [15], [21]). In Section VI, we bound the performance of this algorithm with respect to the optimal, or ML, performance and in Section VII, we analyze the complexity of the proposed algorithm. We then present simulations in Section VIII. Ideas for future work and conclusions are to be found in Section IX.

## II. INTEGER LEAST-SQUARES PROBLEM

The integer least-squares problem is the following minimization problem

$$\min_{s \in \mathbb{Z}^{M \times 1}} \|x - Hs\|^2$$

where  $x \in \mathbb{C}^{N \times 1}$  and  $H \in \mathbb{C}^{N \times M}$  are known, and  $\mathbb{Z}^{M \times 1}$  is the  $M$ -dimensional integer lattice. Often the search space is a finite subset of the integer lattice, say  $\mathcal{A}$ , in which case the minimization is done over  $s \in \mathcal{A}$  rather than  $s \in \mathbb{Z}^{M \times 1}$ . This problem arises in several situations in communications, cryptography, etc. For a general  $H$ , it is known to be NP hard in the worst-case sense [23], as well as the average sense [12], [24]. We now describe this problem in the context of ML decoding in a multiple antenna system.

### A. System Model

We assume a discrete-time block-fading multiple antenna channel model with  $M$  transmit and  $N$  receive antennas, where the channel is known. This is a reasonable assumption for communication systems where the signaling rate is much higher than the rate at which the propagation environment changes, so that the channel may be learned (perhaps by transmitting known training sequences) by the receiver. If  $\mathcal{S}$  is the finite signal constellation, then during any channel use, the transmitted signal  $\tilde{s} \in \mathcal{S}^{M \times 1}$  and the received signal  $x \in \mathbb{C}^{N \times 1}$  are related by

$$x = H\tilde{s} + v \quad (1)$$

where  $H \in \mathbb{C}^{N \times M}$  is the known channel matrix with independent, identically distributed (i.i.d.) complex Gaussian entries of

variance  $\sigma_h^2$ , i.e.,  $\mathcal{CN}(0, \sigma_h^2)$ . We assume  $N \geq M$ ,  $v \in \mathbb{C}^{N \times 1}$  is the unknown additive noise vector, comprised of i.i.d. complex Gaussian entries of variance  $\sigma_v^2$ , i.e.,  $\mathcal{CN}(0, \sigma_v^2)$ . Without loss of generality, we assume  $\sigma_v^2 = 1$ . Thus,  $H$  and  $v$  are the only sources of randomness when  $\tilde{s}$  is a particular transmitted point. With this setup, we have  $\sigma_h = (\sigma_v/\sigma_s)\sqrt{\rho/M}$  where  $\rho$  is the expected signal-to-noise ratio (SNR) and  $\sigma_s^2$  is the average power of the signal constellation  $\mathcal{S}$ . Under the aforementioned assumptions the ML criterion requires us to find  $s \in \mathcal{S}^{M \times 1}$  that minimizes  $\|x - Hs\|^2$ . This is equivalent to the integer least-squares problem mentioned in Section II, where the search space,  $\mathcal{A}$ , *viz.*,  $\mathcal{S}^{M \times 1}$ , is finite but has cardinality exponential in  $M$ .

This is different from the general integer least-squares problem in that  $H$  and  $v$  are random, and, hence, the complexity of solving this problem is also a random variable. Therefore, it is the various moments of the complexity that are of interest to us—we focus on the expected complexity in this paper.

Also, the underlying probability distributions tell us how to prune the search space in order to reduce the complexity of the general integer least-squares problem while maintaining performance close to optimal.

In this paper, we only consider  $L^2$ -QAM constellations with even  $L$ , i.e.,

$$\mathcal{S} = \left\{ a + jb \mid a, b \in \left\{ -\frac{L-1}{2}, -\frac{L-3}{2}, \dots, \frac{L-3}{2}, \frac{L-1}{2} \right\} \right\}. \quad (2)$$

It is then easy to show that  $\sigma_s^2 = (L^2 - 1)/6$ . This gives us  $\sigma_h = \sqrt{(6/(L^2 - 1))(\rho/M)}$ .

Finally, we note that the above description fits a system in which transmissions are uncoded. In the ML decoding of systems involving space-time codes, etc., we also run into the integer least-squares problem [1]–[3]. In this situation, the operational meanings of  $M$ ,  $N$ , and  $H$  may be different since they now involve the coding scheme as well as the physical antennas. For instance,  $M$  and  $N$  would typically be much larger than the actual number of transmit and receive antennas and  $H$  would have entries that are functions of the coding scheme and the channel values (these would not necessarily be i.i.d. entries). The algorithms mentioned in this paper would work for these systems also; however, the analysis of the performance and computational complexity would be different and would vary from system to system. The analysis of the i.i.d. case is complicated as is and would become even more intractable in the correlated case. Therefore, we restrict the analysis to  $H$  matrices with i.i.d. entries. We deal with non-i.i.d. matrices through simulations where we run the proposed decoder on space-time coded systems that lead to an equivalent channel with correlation.

## III. SPHERE DECODER

In this section, we introduce the sphere decoder and also introduce the notation that will be used in the rest of the paper. In sphere decoding, we search only over lattice points that lie in a hypersphere of radius  $r$  around  $x$ , thus reducing the search

<sup>1</sup>The case  $N < M$  can also be dealt with using the approach of this paper. However, since it inevitably requires an exhaustive search over a lattice of dimension  $M - N$ , we shall not consider it here.

space and the computation. Therefore, we first need to find all  $s \in \mathcal{S}^{M \times 1}$  that lie within this hypersphere of radius  $r$ . This is equivalent to solving

$$r^2 \geq \|x - Hs\|^2. \quad (3)$$

To this end, consider the QR decomposition of the channel matrix,  $H = Q \begin{bmatrix} R \\ 0_{(N-M) \times M} \end{bmatrix}$  where  $R$  is an  $M \times M$  upper triangular matrix with non-negative diagonal entries and  $Q$  is an  $N \times N$  unitary matrix. Such a decomposition is unique. Partition  $Q$  as  $[Q_1 \ Q_2]$  where  $Q_1$  is  $N \times M$  and  $Q_2$  is  $N \times (N - M)$ . Since  $Q$  is unitary, so is  $Q^*$ . We know that premultiplying by a unitary matrix does not change the squared-norm of a vector. Therefore, (3) becomes

$$r^2 \geq \|x - Hs\|^2 = \left\| x - Q \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 = \left\| \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} x - \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2. \quad (4)$$

Define

$$z = \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} x - \begin{bmatrix} R \\ 0 \end{bmatrix} s. \quad (5)$$

Introduce  $\lambda$  to denote the mod-squared entries of  $z$

$$\lambda_i = |z_{N-i+1}|^2 \quad \text{for } i = 1, \dots, N.$$

Note that  $\lambda$  is indexed backwards relative to  $z$ . From (4), finding all  $s$  that satisfy (3) amounts to finding all  $s$  that satisfy

$$\lambda_1 + \lambda_2 + \dots + \lambda_N \leq r^2. \quad (6)$$

Consider the lower  $N - M$  entries of  $z$ . These are given by the vector  $Q_2^* x$ . Now,  $x$  is known to the receiver and since it knows  $H$ , it can calculate  $Q$  and  $R$ . Therefore,  $Q_2^* x = [z_{M+1}, \dots, z_N]^T$  is known to the receiver. Hence, so are  $\lambda_1, \dots, \lambda_{N-M}$ . Moreover, these are independent of  $s$  and  $\tilde{s}$  and, therefore, contain no useful information for the decoder. Therefore, solving (6) is equivalent to solving

$$\lambda_{N-M+1} + \dots + \lambda_N \leq r'^2 \quad (7)$$

for  $r'^2 = r^2 - (\lambda_1 + \dots + \lambda_{N-M})$ . Note that due to the upper triangularity of  $R$ ,  $\lambda_{i+N-M}$  depends only on the unknowns  $s_M, \dots, s_{N-i+1}$  for  $i = 1, \dots, M$ . Therefore, (7) can be solved by successively solving

$$\begin{aligned} \lambda_{1+N-M} &\leq r'^2 \\ \lambda_{1+N-M} + \lambda_{2+N-M} &\leq r'^2 \\ &\vdots \\ \lambda_{1+N-M} + \lambda_{2+N-M} + \dots + \lambda_N &\leq r'^2 \end{aligned} \quad (8)$$

for  $s_M, s_{M-1}, \dots, s_1$ . This works in the following way. The first condition gives possible values for  $s_M$ . For each of these, using the second condition, we obtain possible values for  $s_{M-1}$ . This process continues because for any predetermined  $s_M, \dots, s_{M-i+2}$ , the  $i$ th condition gives an interval for  $s_{M-i+1}$ . Once all  $s \in \mathcal{S}^{M \times 1}$  that satisfy (7) are known, we can find that  $s$  which minimizes  $\|x - Hs\|^2$ . If there are no solutions found, we increase  $r'$  and resolve the problem. For more on the sphere decoder, see [13].

#### IV. COMPUTATIONAL COMPLEXITY

Computational complexity is defined as the number of arithmetical operations required before the decoder gives an output. Apart from the complexity of the  $QR$  factorization, the major computation involved in finding the closest point is in determining all points in each lower dimension, i.e., in the successive inequalities of (8). We see that the algorithm constructs a search tree where the branches at depth  $k$  in the tree correspond to the lattice points inside the hypersphere of radius  $r$  and dimension  $k$ . Clearly, the total computation involved depends on the number of points the decoder visits as it constructs the tree. For a point in the  $k$ th dimension, the number of operations or flops required to process it turn out to be proportional to  $k$  ( $2k + 17$  in [13]). Therefore, we have

$$C = \sum_{k=1}^M (\text{Expected \# of points in } k\text{-sphere of radius } r) \cdot (\text{flops/point}). \quad (9)$$

Thus, the complexity of the algorithm depends on the size of the search tree and the computation required at each dimension. For various implementations, the (flops/point) can take different values and have a complicated dependence on the enumeration method especially for hardware implementations [25]. In particular, the pseudocode of [13] and that presented in Section V-C use a number of flops linear in the dimension under consideration. We will see in the analyses presented in this paper that this factor either plays no role (asymptotic analysis of Section VII-B) or remains transparent in the final expression (Sections VII-A). Thus, replacing it by a different expression presents no difficulty as far as the analysis is concerned. As for the simulations presented in this paper, our particular implementation of the algorithm in MATLAB does use flops/point that are linear in the dimension and we use this fact while presenting numerical results.

For the setup involving a real channel and 2-PAM as the signal space and with the receiver using sphere decoding, [13] obtains the following complexity:

$$C = \sum_{k=1}^M (2k + 17) \sum_{l=0}^k \binom{k}{l} \Gamma \left( \frac{r^2}{2(1 + \frac{4\rho}{M}l)}, \frac{k + N - M}{2} \right) \quad (10)$$

where  $\Gamma(x, a) = \int_0^x (e^{-t}/\Gamma(a)) t^{a-1} dt$  is the incomplete gamma function; [13] also has similar expressions for other constellations.

While the sphere decoding algorithm is one of the exact methods that solve the maximum likelihood problem without exhaustive search, even with finite constellations ( $L$ -PAM,  $L^2$ -QAM, etc.), it begins to take up significantly more than  $N^3$  or  $N^4$  computations at some  $N$  which is in the range of practical interest. The reason for this is understood as follows. The chosen radius squared,  $r^2$ , is typically proportional to  $N$ ; therefore, the algorithm retains a very large fraction of the lattice points (in fact, nearly all the points) up to some dimension  $k$  before it starts to prune the tree. For instance, if  $N = 100$ , we have  $r^2 = \alpha N$  such that up to dimension  $k = cN$  where  $c$  is some constant less than 1, we keep nearly all the points of the lattice. This already gives us  $L^{cN}$  points to search over and

TABLE I  
CHARACTERISTIC FUNCTION AND PDF OF  $\lambda_i$

	$Ee^{j\alpha\lambda_i}$	$p_{\lambda_i}(\lambda_i)$
$i \leq N - M$	$\frac{1}{1 - \frac{j\alpha}{c_0}}$	$c_0 e^{-c_0 \lambda_i}$
$i > N - M$	$\frac{(1 - \frac{j\alpha}{c_{i-1-N+M}})^{i-1}}{(1 - \frac{j\alpha}{c_{i-N+M}})^i}$	$\frac{c_{i-N+M}^{i-1}}{c_{i-1-N+M}^{i-1}} e^{-c_{i-N+M} \lambda_i} \sum_{k=0}^{i-1} \binom{i-1}{k} \frac{\lambda_i^k}{k!} (c_{i-1-N+M} - c_{i-N+M})^k$

TABLE II  
MEAN AND VARIANCE OF  $\lambda_i$

	$E\lambda_i$	$\text{var } \lambda_i$
$i \leq N - M$	$\frac{1}{c_0}$	$\frac{1}{c_0^2}$
$i > N - M$	$\frac{i}{c_{i-N+M}} - \frac{(i-1)}{c_{i-1-N+M}}$	$\frac{i}{c_{i-N+M}^2} - \frac{(i-1)}{c_{i-1-N+M}^2}$

the complexity quickly becomes exponential. The result of [14] makes this observation rigorous, and we will discuss this issue further in Section VII-B.

## V. STATISTICAL PRUNING

With a view to decreasing the computational complexity, we now propose a modification to the sphere decoding algorithm that reduces the size of the tree. We suggest the increasing radii algorithm that defines a region around  $x$ , different from the hypersphere, in which to search. This algorithm does not perform exact ML decoding, but can give performance as close to ML as desired through the choice of certain parameters. The proposed algorithm relies heavily on the statistics of the problem (such as the distribution of the  $\lambda_i$ ) for performance, as well as reduction in complexity. In fact, it is the statistics that motivate the particular pruning approach that we take.

### A. Statistics

We now take a look at these statistics. For any vector  $s \in \mathcal{S}^{M \times 1}$ , define  $s^i \in \mathcal{S}^{i \times 1}$  as the lower length- $i$  subvector of  $s$ , i.e., the vector  $[s_{M-i+1}, \dots, s_M]^T$ . Define  $c_i = 1/(\sigma_v^2 + \sigma_h^2 \|s^i - \hat{s}^i\|^2)$  and  $c_0 = 1/\sigma_v^2 = 1$ .

The characteristic functions and distributions for the  $\lambda_i$  random variables are obtained in Appendix Section A and mentioned in Table I. The mean and variance can then be computed easily and are mentioned in Table II.

We note that the  $\lambda_i$ s are independent random variables. Define  $\beta_{i,j} = \sum_{k=i}^j \lambda_{k+N-M}$  for  $1 \leq i \leq j \leq M$ . We denote  $\beta_{1,i}$  by  $\beta_i$ . Thus,  $\beta_i$  is simply the sum of  $i$  independent random variables. Therefore, its characteristic function is the product of the relevant  $\lambda_j$  characteristic functions. Now the statistics for the  $\beta_i$  random variables are easy to compute and are shown in Table III. Note that the  $\beta_i$  are the quantities on the left side of (8).

The sphere decoder gives exponential complexity because the first several conditions of (8) are very loose. Thus, the tree of the points visited grows exponentially for the first several dimensions. This is also clear from the fact that the sums  $\lambda_{1+N-M} + \dots + \lambda_{k+N-M}$  which occur in (8) (viz., the  $\beta_k$ s) have monotonically increasing means while  $r'$  is typically chosen on the basis of the distribution of  $\beta_M$ , i.e., the full sum of all the  $\lambda_i$ s under consideration. Therefore, the first several conditions do

not prune the search space as much as desired. Taking our cue from this, we propose a modification to the sphere decoding algorithm. In this modification, we prune the search space right from the lower dimensions.

### B. Increasing Radii Algorithm (IRA)

Using a schedule of radii  $r_1 \leq r_2 \leq \dots \leq r_M$  we solve for

$$\begin{aligned} \lambda_{1+N-M} &\leq r_1^2 \\ \lambda_{1+N-M} + \lambda_{2+N-M} &\leq r_2^2 \\ &\vdots \\ \lambda_{1+N-M} + \lambda_{2+N-M} + \dots + \lambda_N &\leq r_M^2 \end{aligned} \quad (11)$$

instead of (8). By choosing a smaller radius for the lower dimensions and gradually increasing it, the search space is cut down much earlier than with the sphere decoder. We hope that this will reduce the number of points in the search region at the lower dimensions. Denote by  $\mathcal{D}_k$  the region in  $\mathcal{S}^{k \times 1}$  containing points that satisfy the first  $k$  inequalities of (11) (note that these points have been determined by finding the values of  $s_M, s_{M-1}, \dots, s_{M-k+1}$  that satisfy the first  $k$  conditions). We refer to  $\mathcal{D}_M$  as  $\mathcal{D}$  in the following discussion. As in the sphere decoder we can determine all  $s \in \mathcal{D}$  by solving the inequalities in (11) successively. Once the points within  $\mathcal{D}$  are determined, we find that point in  $\mathcal{D}$  which minimizes  $\|x - Hs\|$  and declare it as the decoder output.

To reduce the complexity, we naturally try to reduce the number of points in  $\mathcal{D}$ . However, because of the ‘‘asymmetry’’ of the region it is possible that the lattice point closest to  $x$  does not lie in the search space. For the sphere decoder, the closest point to  $x$  inside the hypersphere is the closest point to  $x$  in the entire lattice. For the IRA, however, the closest point to  $x$  in  $\mathcal{D}$  is not necessarily the closest point to  $x$  in the entire lattice. Thus, unlike the sphere decoder, we are *not* doing ML decoding and are, potentially, incurring a greater BER. What we get in return is reduced computational complexity. By increasing the asymmetry of the search region we can decrease the computation involved, but simultaneously incur an increased BER. This is the tradeoff inherent in the modification. As with the sphere decoder, if  $\mathcal{D}$  is empty, we increase the search region and run the decoder again. We note in passing that similarly named algorithms are presented in [20]. However, they differ significantly from this method of pruning as they rank most promising paths within a fixed search radius in order to limit computation. This makes them more efficient in some cases but also harder to analyze. The main difference between the pruning of [20] and the approach of this paper is that in the former, the pruning depends on the precise channel realization for that transmission, while

TABLE III  
STATISTICS OF  $\beta_i$ 

	$E^{j\alpha\beta_i}$	$p_{\beta_i}(\beta_i)$	$E\beta_i$	$\text{var } \beta_i$
$1 \leq i \leq M$	$\frac{(1-\frac{2\alpha}{c_0})^{N-M}}{(1-\frac{2\alpha}{c_i})^{i+N-M}}$	$\frac{c_i^{i+N-M}}{c_0^{N-M}} e^{-c_i\beta_{1,i}} \sum_{l=0}^{N-M} \binom{N-M}{l} \frac{\beta_{1,i}^{i+l-1}}{(i+l-1)!} (c_0 - c_i)^l$	$\frac{i+N-M}{c_i} - \frac{N-M}{c_0}$	$\frac{i+N-M}{c_i^2} - \frac{N-M}{c_0^2}$

TABLE IV  
PSEUDOCODE FOR THE INCREASING RADII ALGORITHM

<b>function</b> DECODE( $x, H, r$ )	$x$ : received vector, $H$ : known channel, $r$ : vector of the radii schedule
1. $H = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$	$QR$ decomposition of $H$ . $R$ has a real diagonal
2. $t = Q^*x, y = [t_1, \dots, t_M]$	$y$ has the first $M$ elements of $t$
3. $\mathcal{D} = \emptyset, y'' = r' = s = 0_{M \times 1}$	Initialization: $\mathcal{D}$ as the set of vectors in the search region
4. <b>while</b> $\mathcal{D} = \emptyset$	Repeat till search region is non-empty
$r = \text{GETNEWSCHEDULE}$	Obtain new schedule with smaller $\epsilon$
$\mathcal{D} = \text{DECREASE}(N, y, R, r, y'', r', s, \mathcal{D})$	Call subroutine
5. $s^* = \text{argmin}_{s \in \mathcal{D}} \ x - Hs\ ^2$	Find closest element within search region
6. <b>output</b> $s^*$	Decoder output
<b>function</b> DECREASE( $k, y, R, r, y'', r', s, \mathcal{D}$ )	$k$ : subdimension, $s$ : vector under consideration, $s^{M-k}$ is known.
1. <b>if</b> $k = 0$	Subroutine finds possible values for $s_k$ .
$\mathcal{D} = \mathcal{D} \cup \{s\}$	Check if subdimension is zero
<b>return</b>	Conclude that $s$ is inside the search region
2. <b>elseif</b> $k = N$	At the highest dimension
$r'_k = r_1, y''_k = y_1$	Initialize
3. <b>else</b>	Calculations to find permissible values of $s_k$
$y''_k = y_k - \sum_{j=k+1}^N r_{k,j} s_j$	
$r'_k = \left( (r_{N-k+1}^2 - r_{N-k}^2) + r_{k+1}^2 - (y_{k+1}'' - R_{k+1,k+1} s_{k+1})^2 \right)^{1/2}$	
4. $LB = \max \left( \left\lfloor \frac{r'_k + y''_k}{r_{k,k}} - \frac{1}{2} \right\rfloor + \frac{1}{2}, -\frac{L-1}{2} \right), UB = \min \left( \left\lceil \frac{-r'_k + y''_k}{r_{k,k}} + \frac{1}{2} \right\rceil - \frac{1}{2}, \frac{L-1}{2} \right)$	Exact range of $s_k$ with L-PAM
5. <b>for</b> $n = LB : UB$	For each possible value of $s_k$
$s_k = n$	Assign that value to $s_k$
$\mathcal{D} = \text{DECREASE}(k-1, y, R, r, y'', r', s, \mathcal{D})$	Call subroutine to find possible values of $s_{k-1}$
6. <b>return</b>	

in the latter the pruning depends only on the statistics of the problem (in fact, only the SNR) and does not depend on the actual channel realization.

### C. Pseudocode

The algorithm is in pseudocode in Table IV. It uses a depth-first search to construct the tree. We use the vector  $r$  of size  $M \times 1$  to denote the schedule  $r_1, \dots, r_M$  that we are using for the decoding. GETNEWSCHEDULE returns the new sequence of  $r_i$ s with which we repeat the search when the region  $\mathcal{D}$  is empty. The first schedule is chosen so as to be successful with some probability  $(1 - \epsilon_1)$ . If it fails, the second is chosen so as to be successful with probability  $(1 - \epsilon_2)$ , etc. This will become clearer in later sections. Clearly, for all  $r_i$  being equal, the IRA is the same as the sphere decoder.

## VI. PROBABILITY OF ERROR

The algorithm repeats the search with a new sequence of  $r_i$ s if the solution set of (11), viz.,  $\mathcal{D}$ , is empty. Let  $\mathcal{D}^i$  be the solution set at the  $i$ th iteration. The algorithm terminates at the first  $i$  for which  $\mathcal{D}^i \neq \emptyset$ . We assume that  $\mathcal{D}^{i-1} \subseteq \mathcal{D}^i$  and  $\mathcal{D}^\infty = \mathcal{S}^{M \times 1}$ .

Recall that  $\tilde{s}$  is the transmitted point. Define  $\epsilon_i = P(\tilde{s} \notin \mathcal{D}^i)$ . With probability  $P(\text{error})$  we make an error by decoding

to  $s \neq \tilde{s}$

$$\begin{aligned}
 P(\text{error}) &= \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset) \\
 &= \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \in \mathcal{D}^i) \\
 &\quad + \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i) \\
 &= \sum_{i=1}^{\infty} P\left(\|x - H(s - \tilde{s})\|^2 \leq \|v\|^2 \text{ for } s \in \mathcal{D}^i, \right. \\
 &\quad \left. s \neq \tilde{s}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \in \mathcal{D}^i\right) \\
 &\quad + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i) \\
 &\leq \sum_{i=1}^{\infty} P(\text{ML decoder error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset) \\
 &\quad + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i) \quad (12)
 \end{aligned}$$

$$\begin{aligned}
&= P_e^{ML} + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i) \\
&\leq P_e^{ML} + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^1) \\
&= P_e^{ML} + \epsilon_1
\end{aligned} \tag{13}$$

where  $P_e^{ML}$  is the probability of error with ML decoding. The third equality comes from the fact that an error is certain to be made if  $\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin \mathcal{D}^i$  since the transmitted point is not in  $\mathcal{D}^i$  while some other point is. The first inequality comes from the fact that an ML decoder error does not require  $s$  or  $\tilde{s}$  to be  $\mathcal{D}^i$ . We expect that (12) is a tight bound relating the probability of error of the modified algorithms to  $P_e^{ML}$ . This is because it takes into account all the successive schedules of  $r_i$  that the algorithms may go through. However, it is not clear how to evaluate it exactly, and, hence, we propose the simple bound of (13). This would be equal to (12) if we chose to use only one schedule of  $r_i$  and declared all bits to be in error if the corresponding  $\mathcal{D}$  turned out to be empty, rather than increasing the  $r_i$  and running the decoder again.

#### A. $\epsilon$ With Increasing Radii Algorithm

For any given set of radii  $r_1 \leq \dots \leq r_M$ ,  $\mathcal{D}$  denotes the set of the lattice points inside the search region. We now compute  $\epsilon = P(\tilde{s} \notin \mathcal{D})$  for the increasing radii algorithm.

*Lemma 1:* For the IRA, given a set of radii  $r_1 \leq \dots \leq r_M$ ,  $\epsilon = P(\tilde{s} \notin \mathcal{D})$  is given by

$$\epsilon = \sum_{k=1}^M e^{-r_k^2} J_{k-1} \tag{14}$$

where

$$J_k = \sum_{l=0}^{k-1} (-1)^{k-l+1} \frac{r_{l+1}^{2(k-l)}}{(k-l)!} J_l, \quad J_0 = 1. \tag{15}$$

*Proof:* If  $s = \tilde{s}$ , we have  $z = Q^*v$ . Since  $Q$  is unitary,  $Q^*v$  has the same statistics as  $v$ , i.e., i.i.d. entries distributed as  $\mathcal{CN}(0, 1)$ . With  $\lambda_i = |z_{N-i+1}|^2$ , we have  $p_{\lambda_i}(\lambda_i) = e^{-\lambda_i} \cdot 1 - \epsilon$  is the probability that  $\lambda_{1+N-M}, \dots, \lambda_N$  satisfy (11). Because the  $\lambda_i$ s are independent

$$\begin{aligned}
&p_{\lambda_{1+N-M}, \lambda_{2+N-M}, \dots, \lambda_N}(\lambda_{1+N-M}, \lambda_{2+N-M}, \dots, \lambda_N) \\
&= e^{-(\lambda_{1+N-M} + \lambda_{2+N-M} + \dots + \lambda_N)}.
\end{aligned}$$

Therefore, see the equation at the bottom of the page, where the second line comes from changing variables

$\mu_i = \sum_{j=1}^i \lambda_{j+N-M}$  for  $i = 1, \dots, M$ . If we call this integral  $I_M$  and integrate out  $\mu_M$ , we get

$$I_M = I_{M-1} - e^{-r_M^2} J_{M-1} \tag{16}$$

where  $J_{M-1} = \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \dots \int_{\mu_{M-2}}^{r_{M-1}^2} d\mu_{M-1} \dots d\mu_1$ . It can be shown that the  $J_i$ s satisfy the recurrence of (15). Thus,  $J_0, \dots, J_{M-1}$  can be computed. We define  $I_0 = 1$ . Then, using (16) recursively, we get  $I_M = 1 - \sum_{k=1}^M e^{-r_k^2} J_{k-1}$ . Since  $1 - \epsilon = I_M$ , we get (14).  $\square$

#### B. Choice of $\epsilon$ and the Radii

Thus, we obtain an exact expression for  $\epsilon$ . Once we decide how much worse than ML we are prepared to be, we can choose  $\epsilon$  using the bound in (13). As indicated earlier, this bound is loose, and the performance is usually much better than that indicated by the value of  $\epsilon$ . For the chosen value of  $\epsilon$ , we can then use the expressions above to determine the radii  $r_1, \dots, r_M$ . Note, however, that since (14) gives a highly underdetermined equation system involving the  $r_i$ s there is an entire family of schedules of  $r_i$  that give a particular epsilon. However, if we choose a functional form for the radii, we can use the expressions obtained above to determine the  $r_i$ s. Since we want to solve (11), choosing the  $r_i$ s in accordance with the expected values of the partial sums that appear on the left side of each inequality is a reasonable option. However, these partial sums are precisely the  $\beta_i$ s. The statistics of these are in Table III. We can see that their expected values are  $((i + N - M)/c_i) - ((N - M)/c_0) = (i + N - M)(\sigma_v^2 + \sigma_h^2 \|s^i - \tilde{s}^i\|^2) - (N - M)\sigma_v^2$ . Although  $\|s^i - \tilde{s}^i\|^2$  can take a range of values, we can see that  $E/\beta_i$  increases at least linearly with  $i$ . This motivates us to settle upon a linear schedule for the  $r_i^2$ s. This also means we have fewer parameters to choose. As indicated in the calculation of  $\epsilon$ , the  $r_i^2$  values are chosen with the noise statistics in mind; therefore, the slope of linearity is chosen as  $\sigma_v^2$  (this is typically 1). It is now enough to choose the value of  $r_1^2$  to determine the entire schedule. If we choose  $r_1^2 = (\delta \log M + 1)\sigma_v^2$ , then the probability that the transmitted signal falls outside the search region at the first dimension decays as  $1/eM^\delta$ . Therefore, we set  $r_i^2 = (\delta \log M + i)\sigma_v^2$ , and choose  $\delta$  such that  $\epsilon = 0.01$ , etc. Thus, we can stay as close to the ML performance as we desire through choice of  $r_i$ s.

In Table V, we list some values of  $\delta$  for different values of  $M$ . This means that if we desire a value of  $\epsilon$  for a particular value of  $M$ , a radius schedule of  $r_i^2 = (\delta \log M + i)\sigma_v^2$  where  $\delta$  is picked from the table will do the job.

$$\begin{aligned}
1 - \epsilon &= \int_0^{r_1^2} \int_0^{r_2^2 - \lambda_{1+N-M}} \dots \int_0^{r_M^2 - (\lambda_{1+N-M} + \dots + \lambda_{N-1})} e^{-(\lambda_{1+N-M} + \dots + \lambda_N)} d\lambda_N \dots d\lambda_{1+N-M} \\
&= \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \dots \int_{\mu_{M-1}}^{r_M^2} e^{-\mu_M} d\mu_M \dots d\mu_1
\end{aligned}$$

TABLE V  
VALUES OF  $\delta$  FOR VARIOUS VALUES OF  $M$  AND  $\epsilon$ . FOR A PAIR OF  
VALUES  $M$  AND  $\epsilon$ , USE THE CORRESPONDING VALUE OF  $\delta$  FROM  
THE TABLE AND A SCHEDULE OF  $r_i^2 = (\delta \log M + i)\sigma_v^2$

$\delta$	$M = 10$	$M = 20$	$M = 30$	$M = 40$	$M = 50$
$\epsilon = 0.1$	2.16	2.35	2.55	2.74	2.93
$\epsilon = 0.01$	4.09	4.29	4.48	4.67	4.96
$\epsilon = 0.001$	5.64	5.83	6.03	6.41	6.61
$\epsilon = 0.0001$	7.19	7.19	7.48	7.77	8.15

## VII. ANALYSIS OF COMPUTATIONAL COMPLEXITY

Recall the concept of computational complexity outlined in Section (4). In particular, we focus on the expression in (9). Since we are not searching over hyperspheres any more we have a modified expression for the complexity

$$C = \sum_{k=1}^M (\text{Expected \# of points in } \mathcal{D}_k) \cdot (\text{flops/point}). \quad (17)$$

From the pseudocode of Section V-C, we can determine that the flops/point is  $8k + 32$ .

Let us now investigate the exact computational complexity as defined in (17).  $s^k$  is as defined in Section V-A. Define  $P(s^k \in \mathcal{D}_k)$  to be the probability that the point  $s^k$  is in the search region at dimension  $k$ , i.e., it satisfies the first  $k$  equations of (11). Clearly

$$\text{Expected \# of points in } \mathcal{D}_k = \sum_{s^k \in \mathcal{S}^{k \times 1}} P(s^k \in \mathcal{D}_k). \quad (18)$$

We now need to compute  $P(s^k \in \mathcal{D}_k)$  and then do the sum in (18). Note that the number of terms in the sum is  $L^{2k}$ , i.e., exponential in  $k$ . Naturally, we would like to evaluate the sum *without* having to explicitly evaluate  $P(s^k \in \mathcal{D}_k)$  for each of the  $L^{2k}$  values of  $s^k$ . Whether this can be done or not depends on the functional form of  $P(s^k \in \mathcal{D}_k)$ . Therefore, while determining  $P(s^k \in \mathcal{D}_k)$  we also keep in mind the summation of (18).

For any  $s^k \in \mathcal{S}^{k \times 1}$ , the joint distribution of  $\lambda_{1+N-M}, \dots, \lambda_{k+N-M}$  determines  $P(s^k \in \mathcal{D}_k)$ . More specifically, see (19), shown at the bottom of the page.

We know the distribution of the  $\lambda_i$ s from Table (1). Since the  $\lambda_i$ s are independent, we have

$$\begin{aligned} P_{\lambda_{1+N-M}, \dots, \lambda_{k+N-M}}(\lambda_{1+N-M}, \dots, \lambda_{k+N-M}) \\ = \prod_{i=1}^k p_{\lambda_{i+N-M}}(\lambda_{i+N-M}). \end{aligned} \quad (20)$$

Substituting from Table I and (20) into (19), the integral for  $P(s^k \in \mathcal{D}_k)$  can be obtained exactly. However, this integral is

very involved, and, moreover, even if evaluated exactly, would not give an expression that can be summed easily in (18). Ways of approximating this integral and, therefore, the complexity are presented in a technical report [26]. Since this analysis is quite complicated we do not present it in this paper. Instead, we present an upper bound to  $P(s^k \in \mathcal{D}_k)$  and, hence, the complexity. We will also present an asymptotic analysis of this upper bound for large dimensions.

### A. A Simple Upper Bound

We upperbound the number of points in the search region at dimension  $k$  by ignoring the fact that pruning has been done in dimensions less than  $k$ . This means that instead of imposing the first  $k$  conditions of (11) for a point to be in the search region at the  $k$ th subdimension, we only impose the  $k$ th condition. This becomes clearer in the proof of the following result.

*Theorem 1:* For the increasing radii algorithm, the computational complexity is bounded as

$$\begin{aligned} C \leq \sum_{k=1}^M (8k + 32) \sum_{n=0}^{2k(L-1)^2} G_{L,k}[n] \\ \times \sum_{l=0}^{N-M} \binom{N-M}{l} \left( \frac{1}{\sigma_v^2} - \frac{1}{\sigma_v^2 + \sigma_h^2 n} \right)^l \sigma_v^{2(N-M)} \\ \times (\sigma_v^2 + \sigma_h^2 n)^{l-N+M} \Gamma \left( \frac{r_k^2}{\sigma_v^2 + \sigma_h^2 n}, k + l \right) \end{aligned} \quad (21)$$

where  $G_{L,k}[n]$  is the coefficient of  $x^n$  in  $(1/L^{2k})(L + \sum_{j=1}^{L-1} 2(L-j)x^j)^{2k}$  and  $\Gamma(x, a) = \int_0^x (e^{-t}/\Gamma(a))t^{a-1}dt$

*Proof:* Recall that  $\beta_{i,j} = \sum_{k=i}^j \lambda_{k+N-M}$ . For any  $s$ , let  $B_i$  be the event that  $\beta_{1,i} \leq r_i^2$  for  $i = 1, \dots, M$ . The statistics of the  $\beta_i$ s are mentioned in Table III.  $s^k \in \mathcal{D}_k$  if it satisfies the first  $k$  conditions of (11). This happens with probability  $P(B_1, \dots, B_k)$ . Now, if we only wanted to impose the  $k$ th condition, it would be satisfied with probability  $P(B_k)$ . Naturally,  $P(B_k)$  upperbounds  $P(B_1, \dots, B_k)$ . Therefore

$$\begin{aligned} P(s^k \in \mathcal{D}_k) &= P(B_1, \dots, B_k) \\ &\leq P(B_k) \\ &= \int_0^{r_k^2} p_{\beta_{1,k}}(\beta_{1,k}) d\beta_{1,k} \\ &= \sum_{l=0}^{N-M} \binom{N-M}{l} \frac{(c_0 - c_k)^l}{c_0^{N-M}} c_i^{N-M-l} \Gamma(c_k r_k^2, k + l) \\ &= \sum_{l=0}^{N-M} \binom{N-M}{l} \left( \frac{1}{\sigma_v^2} - \frac{1}{\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2} \right)^l \\ &\quad \times \sigma_v^{2(N-M)} (\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2)^{l-N+M} \\ &\quad \times \Gamma \left( \frac{r_k^2}{\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2}, k + l \right) \end{aligned}$$

---


$$\begin{aligned} P(s^k \in \mathcal{D}_k) = \int_0^{r_1^2} \dots \int_0^{r_k^2 - (\lambda_{1+N-M} + \dots + \lambda_{k-1+N-M})} \\ p_{\lambda_{1+N-M}, \dots, \lambda_{k+N-M}} \\ \times (\lambda_{1+N-M}, \dots, \lambda_{k+N-M}) d\lambda_{k+N-M} \dots d\lambda_{1+N-M} \end{aligned} \quad (19)$$

where  $\Gamma(x, a) = \int_0^x (e^{-t}/\Gamma(a))t^{a-1}dt$  is the incomplete gamma function.

We now need to evaluate the summation of (18) with this upper bound. From the definition of  $\mathcal{S}$  in (2), it is evident that each entry in  $s^k - \tilde{s}^k$  can only take values of the form  $x + jy$  where  $x, y \in \{-(L-1), -(L-2), \dots, (L-2), (L-1)\}$ . Therefore,  $\|s^k - \tilde{s}^k\|^2$  can take values in  $\{0, \dots, 2k(L-1)^2\}$ . Denote by  $r_k^L(n)$  the “average” number of solutions to  $\|s^k - \tilde{s}^k\|^2 = n$ . More precisely

$$r_k^L(n) = \frac{1}{L^{2k}} \sum_{\tilde{s}^k \in \mathcal{S}^{k \times 1}} (\text{number of } s^k \in \mathcal{S}^{k \times 1} \text{ such that } \|s^k - \tilde{s}^k\|^2 = n). \quad (22)$$

We have assumed, without loss of generality, that all points are equally likely to be transmitted. With this the summation of (18) becomes

$$\begin{aligned} & \sum_{s^k \in \mathcal{S}^{k \times 1}} P(s^k \in \mathcal{D}_k) \\ &= \sum_{n=0}^{2k(L-1)^2} r_k^L(n) \sum_{l=0}^{N-M} \binom{N-M}{l} \left( \frac{1}{\sigma_v^2} - \frac{1}{\sigma_v^2 + \sigma_h^2 n} \right)^l \sigma_v^{2(N-M)} \\ & \quad \times (\sigma_v^2 + \sigma_h^2 n)^{l-N+M} \Gamma\left(\frac{r_k^2}{\sigma_v^2 + \sigma_h^2 n}, k+l\right). \end{aligned} \quad (23)$$

It is shown in Appendix Section B that  $r_k^L(n)$  is given by the coefficient of  $x^n$  in  $G_L^k(x)$  where  $G_L(x)$  is the generating function mentioned in the statement of Theorem 1. We denote  $G_L^k(x)$  by  $G_{L,k}(x)$  and the coefficient of  $x^n$  in this by  $G_{L,k}[n]$ . This gives us  $r_k^L(n) = G_{L,k}[n]$ . Using this in (23) and the expressions relating to complexity stated in (17) and (18), we get the upper bound in (21).  $\square$

This upper bound is very easy to evaluate especially for small and moderate values of  $M$  and  $N$ . It is also quite tight in this region. We further note that for  $N = M$ , the upper bound of (21) simplifies to

$$C \leq \sum_{k=1}^M (8k+32) \sum_{n=0}^{2k(L-1)^2} G_{L,k}[n] \Gamma\left(\frac{r_k^2}{\sigma_v^2 + \sigma_h^2 n}, k\right). \quad (24)$$

We also note that for the 4-QAM constellation,  $L = 2$  and  $G_{2,k}[n] = \binom{2k}{n}$ .

The upperbound of this section is valid for all values of  $M$ ,  $N$ ,  $L$ , and SNR. In the following section, we fix  $M = N$  and analyze this upper bound for a fixed SNR and asymptotically large  $N$ .

### B. Asymptotics of the Upper Bound

In this section, we will compare the asymptotic complexities of the sphere decoder and the upper bound on the increasing radii algorithm using some simple arguments. We will assume  $M = N$  and that  $N$  is very large. Let  $r^2 = N$  for the sphere decoder and  $r_i^2 = i$  for the IRA (it turns out that having  $r^2 = N + \delta \log N$  or  $r_i^2 = i + \delta \log N$  for constant  $\delta$  does not affect the asymptotic analysis). The subscripts SD and IR will be used when we discuss the complexities of the sphere decoder and the IRA respectively. Although the analysis can be done for a generic QAM constellation, we only present results for 4-QAM.

This is because the expression for  $G_{L,k}[n]$  in the upperbound of the previous section is a simple binomial coefficient for this case and is more complicated in the generic case.

Consider the complexity expression for the sphere decoder for the case of  $\mathcal{S}$  being the 4-QAM constellation. This is similar to that for the 2-PAM constellation given in (10) except for the fact that at subdimension  $k$ , we are dealing with complex vectors of length  $k$  or real vectors of length  $2k$  (this issue is addressed in [13]). We have the following expression:

$$C_{\text{SD}} = \sum_{k=1}^N (8k+32) \sum_{l=0}^{2k} \binom{2k}{l} \Gamma\left(\frac{r^2}{1 + \frac{2\rho}{N}l}, k\right) \quad (25)$$

where  $\Gamma(x, a) = \int_0^x (e^{-t}/\Gamma(a))t^{a-1}dt$ . From (24), and since  $G_{2,k}[n] = \binom{2k}{n}$ , we have

$$C_{\text{IR}} \leq U_{\text{IR}} = \sum_{k=1}^N (8k+32) \sum_{l=0}^{2k} \binom{2k}{l} \Gamma\left(\frac{r_k^2}{1 + \frac{2\rho}{N}l}, k\right). \quad (26)$$

Note that the only difference between (25) and (26) is that, within the incomplete Gamma function, the  $r^2$  of the former is replaced by  $r_k^2$  in the latter. We now compare  $C_{\text{SD}}$  and  $U_{\text{IR}}$ . Consider the following upper and lower bounds. Both expressions have  $N(N+1)$  terms and the maximum value for  $(8k+32)$  is  $(8N+32)$ . For large  $N$  we have,  $N(N+1)(8N+32) \leq 9N^3$ . Therefore

$$\begin{aligned} & \max_{k=1, \dots, N; l=0, \dots, 2k} \binom{2k}{l} \Gamma\left(\frac{r^2}{1 + \frac{2\rho}{N}l}, k\right) \leq C_{\text{SD}} \\ & \leq 9N^3 \max_{k=1, \dots, N; l=0, \dots, 2k} \binom{2k}{l} \Gamma\left(\frac{r^2}{1 + \frac{2\rho}{N}l}, k\right) \end{aligned}$$

and

$$\begin{aligned} & \max_{k=1, \dots, N; l=0, \dots, 2k} \binom{2k}{l} \Gamma\left(\frac{r_k^2}{1 + \frac{2\rho}{N}l}, k\right) \leq U_{\text{IR}} \\ & \leq 9N^3 \max_{k=1, \dots, N; l=0, \dots, 2k} \binom{2k}{l} \Gamma\left(\frac{r_k^2}{1 + \frac{2\rho}{N}l}, k\right). \end{aligned}$$

It is easy to show that

$$\begin{aligned} \Gamma(x, a) &= e^{-x} \sum_{l=a}^{\infty} \frac{x^l}{l!} \geq e^{-x} \frac{x^a}{a!} \geq e^{-x} \frac{x^a}{\sqrt{2\pi a e} \left(\frac{a}{e}\right)^a} \\ &= \frac{e^{a-x}}{\sqrt{2\pi a e}} \left(\frac{x}{a}\right)^a \end{aligned}$$

where the second inequality comes from Stirling's approximation for large  $a$ :  $a! \leq \sqrt{2\pi a e} \left(\frac{a}{e}\right)^a$ . With this and since  $k \leq N$ , we have

$$\begin{aligned} \Gamma\left(\frac{\nu}{1 + \frac{2\rho}{N}l}, k\right) &\geq \frac{1}{\sqrt{2\pi k e}} \frac{e^{k - \frac{\nu}{1 + \frac{2\rho}{N}l}}}{\left(\frac{k}{\nu} \left(1 + \frac{2\rho}{N}l\right)\right)^k} \\ &\geq \frac{1}{\sqrt{2\pi N e}} \frac{e^{k - \frac{\nu}{1 + \frac{2\rho}{N}l}}}{\left(\frac{k}{\nu} \left(1 + \frac{2\rho}{N}l\right)\right)^k}. \end{aligned}$$



Now, if we upper bound  $\Gamma(\nu/(1 + (2\rho/N)l), k)$  using a simple Chernoff bound, we get

$$\Gamma\left(\frac{\nu}{1 + \frac{2\rho l}{N}}, k\right) \leq \frac{e^{k - \frac{\nu}{1 + \frac{2\rho l}{N}}}}{\left(\frac{k}{\nu} \left(1 + \frac{2\rho l}{N}\right)\right)^k} \quad \text{for } k \geq \frac{\nu}{1 + \frac{2\rho l}{N}}.$$

Note that the upper and lower bounds shown above differ only in the factor of  $1/\sqrt{2\pi N e}$ .

Assume that  $k = bN$  and  $l = aN$  for constants  $a$  and  $b$ . Then  $0 \leq b \leq 1$ ,  $0 \leq a \leq 2b$ . (The condition  $k \geq (\nu/(1 + (2\rho/N)l))$  is always satisfied for the IRA since  $\nu = r_k^2 = k$ . For the sphere decoder,  $\nu = r_k^2 = N$  and the condition translates to  $b \geq (1/(1 + 2\rho a))$ ). The term  $\binom{2k}{l}$  then becomes  $\binom{2bN}{aN}$  and is equal to  $\exp(2bNH(a/2b))$  to the first order in the exponent. (The sequences  $A_N$  and  $B_N$  are said to be equal to the first order in the exponent if  $\lim_{N \rightarrow \infty} \log(a_N/b_N) = 0$ . See [27]. We denote this as  $a_N \doteq b_N$ ). Here,  $H(p) = -p \log p - (1-p) \log(1-p)$  is the entropy function. This gives

$$\begin{aligned} & \binom{2k}{l} \frac{e^{k - \frac{\nu}{1 + \frac{2\rho l}{N}}}}{\left(\frac{k}{\nu} \left(1 + \frac{2\rho l}{N}\right)\right)^k} \\ & \doteq \exp\left\{N\left(2bH\left(\frac{a}{2b}\right) + b - \frac{1}{1 + 2\rho a} - b \log(b(1 + 2\rho a))\right)\right\} \\ & = \exp(N\gamma_{\text{SD}}(a, b)) \end{aligned}$$

where we define  $\gamma_{\text{SD}}(a, b) = 2bH(\frac{a}{2b}) + b - (1/(1 + 2\rho a)) - b \log(b(1 + 2\rho a))$ . Also

$$\begin{aligned} & \binom{2k}{l} \frac{e^{k - \frac{\nu}{1 + \frac{2\rho l}{N}}}}{\left(\frac{k}{\nu} \left(1 + \frac{2\rho l}{N}\right)\right)^k} \\ & \doteq \exp\left\{N\left(2bH\left(\frac{a}{2b}\right) + \frac{2\rho ab}{1 + 2\rho a} - b \log(1 + 2\rho a)\right)\right\} \\ & = \exp(N\gamma_{\text{IR}}(a, b)). \end{aligned}$$

where we define  $\gamma_{\text{IR}}(a, b) = 2bH(\frac{a}{2b}) + (2\rho ab/(1 + 2\rho a)) - b \log(1 + 2\rho a)$ . Thus, the bounds for  $C_{\text{SD}}$  become

$$\begin{aligned} & \frac{1}{\sqrt{2\pi N e}} \max_{0 \leq b \leq 1, 0 \leq a \leq 2b, b \geq \frac{1}{1 + 2\rho a}} \exp(N\gamma_{\text{SD}}(a, b)) \leq C_{\text{SD}} \\ & \leq 9N^3 \max_{0 \leq b \leq 1, 0 \leq a \leq 2b, b \geq \frac{1}{1 + 2\rho a}} \exp(N\gamma_{\text{SD}}(a, b)) \end{aligned}$$

and the bounds on  $U_{\text{IR}}$  become

$$\begin{aligned} & \frac{1}{\sqrt{2\pi N e}} \max_{0 \leq b \leq 1, 0 \leq a \leq 2b} \exp(N\gamma_{\text{IR}}(a, b)) \leq U_{\text{IR}} \\ & \leq 9N^3 \max_{0 \leq b \leq 1, 0 \leq a \leq 2b} \exp(N\gamma_{\text{IR}}(a, b)). \end{aligned}$$

It is easy to check that there are values of  $a$  and  $b$  for which  $\gamma_{\text{SD}}(a, b)$  and  $\gamma_{\text{IR}}(a, b)$  are positive, thus giving exponential bounds on the complexity. Therefore, the terms  $1/\sqrt{2\pi N e}$  and  $9N^3$  are asymptotically insignificant. Thus, the upper and lower bounds match and we have the exact asymptotic behavior. If we denote the asymptotic complexity of the sphere decoder,  $C_{\text{SD}}$ ,

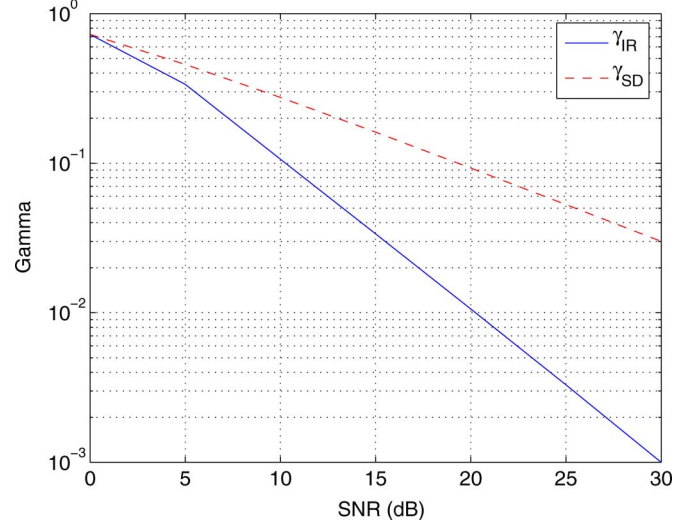


Fig. 1. For large  $N$ , the complexities of the sphere decoder and the IRA are given by  $e^{\gamma_{\text{SD}}N}$  and  $e^{\gamma_{\text{IR}}N}$ , respectively, where  $\gamma$  is as plotted. At 20 dB,  $\gamma_{\text{SD}}$  is roughly ten times  $\gamma_{\text{IR}}$ .

by  $e^{\gamma_{\text{SD}}N}$  and the asymptotic behavior of the upperbound on the IRA complexity,  $U_{\text{IR}}$  by  $e^{\gamma_{\text{IR}}N}$ , we get

$$\begin{aligned} \gamma_{\text{SD}} &= \max_{0 \leq b \leq 1, 0 \leq a \leq 2b, b \geq \frac{1}{1 + 2\rho a}} \gamma_{\text{SD}}(a, b) \\ &= \max_{0 \leq b \leq 1, 0 \leq a \leq 2b, b \geq \frac{1}{1 + 2\rho a}} 2bH\left(\frac{a}{2b}\right) \\ & \quad + b - \frac{1}{1 + 2\rho a} - b \log(b(1 + 2\rho a)) \end{aligned}$$

and

$$\begin{aligned} \gamma_{\text{IR}} &= \max_{0 \leq b \leq 1, 0 \leq a \leq 2b} \gamma_{\text{IR}}(a, b) \\ &= \max_{0 \leq b \leq 1, 0 \leq a \leq 2b} 2bH\left(\frac{a}{2b}\right) + \frac{2\rho ab}{1 + 2\rho a} - b \log(1 + 2\rho a). \end{aligned}$$

Both maximizations are easy to perform numerically. In Fig. 1, we plot the gamma values obtained from the maximizations for different SNRs. Not surprisingly,  $\gamma_{\text{IR}}$  is much lower than  $\gamma_{\text{SD}}$ . This means that the upperbound on the IRA is much lower than the complexity of the sphere decoder. This implies that the actual complexity of the IRA will be even lesser compared to the complexity of the sphere decoder.

We note in passing that, although the large deviations approach of [14] is quite different, it gives exactly the same numerical results as the maximization for  $\gamma_{\text{SD}}$  above. Furthermore, using a similar large deviations approach for the asymptotic analysis of (26) leads to the same  $\gamma_{\text{IR}}$  as above.

## VIII. SIMULATIONS

In this section, we present the results of simulations for different systems. Numerical results for the i.i.d. systems analyzed in the paper are presented as are simulations for a linear dispersion code. In all examples, we have  $M = N$ . We present a comparison of symbol error rates and complexities for the sphere decoder (with Schnorr–Euchner) and that IRA, for different QAM

constellations and values of  $N$  and SNR. Both the sphere decoder and the IRA are run using a depth-first search. For the sphere decoder, we also update the radius once a data point is found inside the sphere. For the IRA, since we use a schedule of radii, rather than a single radius, we do not do any updates.

We note that since  $H$  and  $S$  are complex this amounts to solving  $2N$ -dimensional real problems. The computational complexity  $C$  is presented through the complexity exponent  $C_E = \log C / \log N$ . With this, a complexity exponent of  $C_E$  means that the complexity is  $N^{C_E}$  (clearly,  $C_E$  is different from the  $\gamma$  of Section VII-B).

In all simulations, for the sphere decoder we have used a value of  $r$  chosen to give a particular  $\epsilon$ . For the increasing radii algorithm, we have used a linear schedule of radii, i.e., we have  $r_i = i + \delta \log N$  where  $\delta$  is chosen with some value of  $\epsilon$  in mind. The sequence of  $\epsilon_i$ s that we use is simply 0.1, 0.01, 0.001, etc. This means that we first find  $r$  for the sphere decoder ( $\delta$  for the IRA) which ensures that the transmitted vector is not in the search region with a probability of 0.1 and run the algorithm. If the search region is empty we find a new value of  $r$  ( $\delta$  for the IRA) that gives an  $\epsilon$  of 0.01 and run the algorithm again. This continues till we find a nonempty search region.

Once we have at least one point in the search region, we find, from among those, that point  $s$  which minimizes  $\|x - Hs\|^2$ .

The expression in (17) is used to compute complexity where the (expected # of points in  $\mathcal{D}_k$ ) is estimated by running the decoder on many random instantiations of the problem.  $(8k + 32)$  is the flops/point.

#### A. Computational Complexity and BER

In Figs. 2–4, we look at the complexity exponent and symbol error rate (SER) against the SNR for different values of  $N$  and constellation size.

In Fig. 2, we have  $N = 50$  and  $L = 2$ , which is the 4-QAM constellation. The SNR ranges from 10 to 14 dB. In Fig. 2(a), we see that the complexity exponent can be reduced significantly by using the IRA. We see a complexity that is up to 1.4 orders of magnitude smaller, which means that the IRA can run up to  $50^{1.4} = 240$  times faster. In Fig. 2(b), we see the SER for the IRA. Unfortunately, we have not been able to produce the SER plot for the sphere decoder for this dimension since it would take too long to obtain accurate values.

For the BER comparison, we present results of a smaller sized problem, viz.,  $N = 20$  in Fig. 3. From Fig. 3(a) and (b), we see that with computational savings of 0.8 orders of magnitude (11 times less computation), we get a SER that is very close to the optimal SER ensured by ML decoding.

In Fig. 4, we use  $N = 12$  and  $L = 8$ . This corresponds to a 64-QAM constellation. From Fig. 4(a), we see that the IRA runs around seven times faster than the traditional sphere decoder. From the SER curves of Fig. 4(b) we see that there is no loss of performance.

Not surprisingly, the savings from the IRA are more significant for large  $N$  (this will be further demonstrated in a later simulation). In fact, for systems of dimension 6 and lower we find that the gains relative to the sphere decoder are minimal. This is because the pruning affects fewer dimensions and the

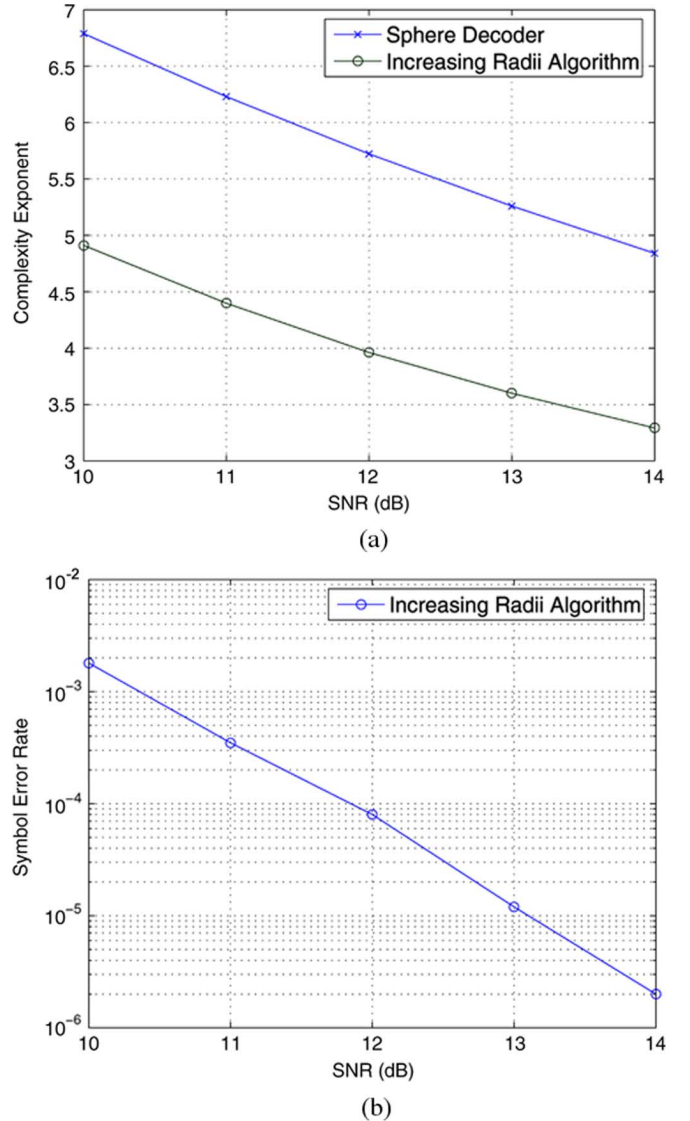


Fig. 2. Complexity exponent and SER for  $M = N = 50$  and 4-QAM. From (a), we see that the IRA can be up to  $50^{1.4} = 240$  times faster than the sphere decoder; (b) shows the symbol error rate with the IRA. (a) Complexity exponent versus SNR; (b) SER versus SNR.

overall complexity is unaffected. Another observation to make from the above set of plots is that (13) is a loose bound since for this setup it says that the proposed algorithms can give SERs that are as much as 0.1 above the optimal. The simulations indicate that this is a gross over-estimate.

#### B. Decoding in a Space-Time Coded System

In this section, we consider the decoding of a system where the equivalent channel is given by a correlated  $H$  matrix rather than an i.i.d. one. Such systems arise commonly in space-time coded systems. We consider the linear dispersion code with eight transmit and four receive antennas with  $T = 8$ ,  $Q = 32$ , and  $R = 16$  presented in [2]. The constellation used is 16-QAM. The equivalent channel used for decoding is a matrix of size  $32 \times 32$  with correlated complex entries. Thus, the decoder works on a real system of dimension 64. In Fig. 5, we present curves for the complexity exponents and the symbol error rates for the

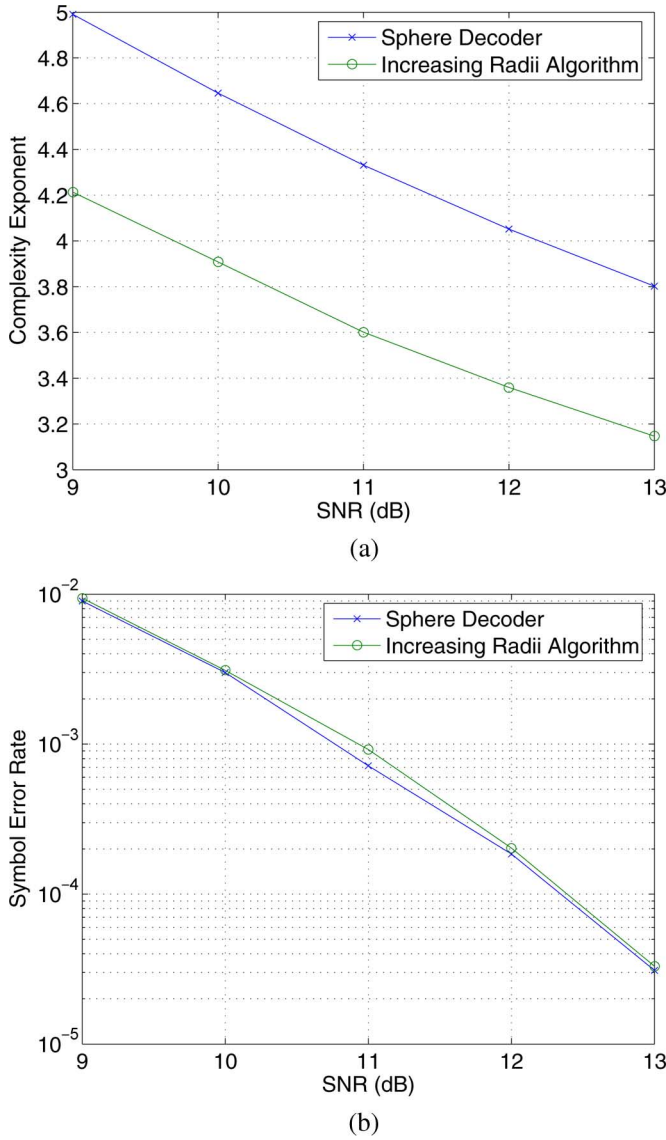


Fig. 3. Complexity Exponent and SER for  $M = N = 20$  and 4-QAM. From (a), we see that the IRA can be up to 11 times faster than the sphere decoder. From (b), we see that the symbol error rates for the two algorithms are very close to each other, indicating no loss of performance. (a) Complexity exponent versus SNR; (b) SER versus SNR.

sphere decoder (with Schnorr–Euchner) and the IRA. We see that the IRA is around 50 times faster and shows almost no loss in performance. Thus, the IRA presents a significant complexity savings while operating in space-time coded systems.

Simulations for the smaller LD code in [2] with four transmit and two receive antennas and  $T = 6$ ,  $R = 8$ ,  $Q = 4$ , and 16-QAM were also done. This gave an equivalent channel of size  $12 \times 12$ . For this, the IRA ran roughly twice as fast as the sphere decoder with an identical symbol error rate in the SNR range of 15 to 25 dB.

### C. Comparing Complexities

From the previous section, it is clear that the IRA can be used to give complexities that are much lower than that of the sphere decoder while still giving BERs close to optimal. Therefore, in

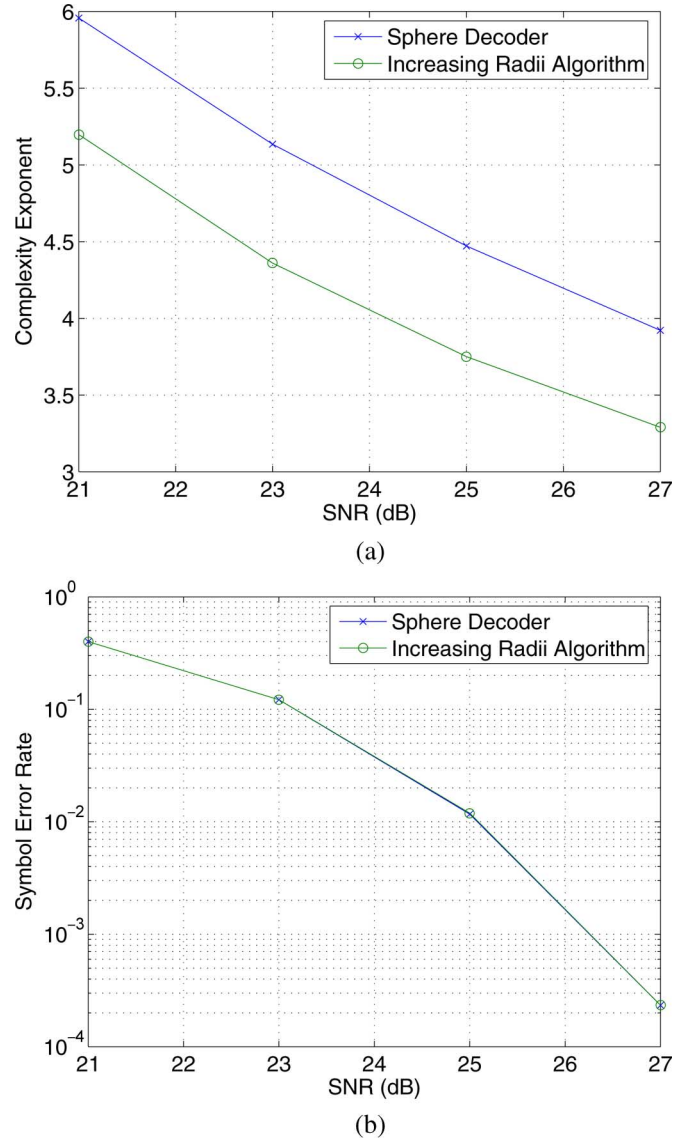


Fig. 4. Complexity Exponent and BER for  $M = N = 12$  and 64-QAM. From (a), we see that the IRA can be up to seven times faster than the sphere decoder. From (b), we see that the symbol error rates for the two algorithms are very close to each other, indicating no loss of performance. (a) Complexity exponent versus SNR; (b) SER versus SNR.

this section, we only compare the complexities of the sphere decoder and the IRA.

In Fig. 6, we compare the complexity of the sphere decoder with that of the IRA in two different ways. In Fig. 6(a), we set the SNR at 27 dB and  $L = 4$ , i.e., a 16-QAM constellation. We vary  $N$  from 20 to 55 and get estimates of the complexity by running the two algorithms sufficiently many times. We see that the complexity exponent of the sphere decoder is increasing rapidly while that of the IRA increases much more slowly. This bears out the analysis of Section VII-B nicely.

In Fig. 6(b), we set  $N = 50$  and  $L = 2$  (4-QAM constellation) and vary the SNR from 10 to 30 dB. We see that the IRA consistently gives us a computational advantage, however, as the SNR increases, both decoders are quite fast and the relative advantage of the IRA diminishes. In particular, at 10 dB, we see that the IRA is around  $50^{1.5} = 300$  times faster.

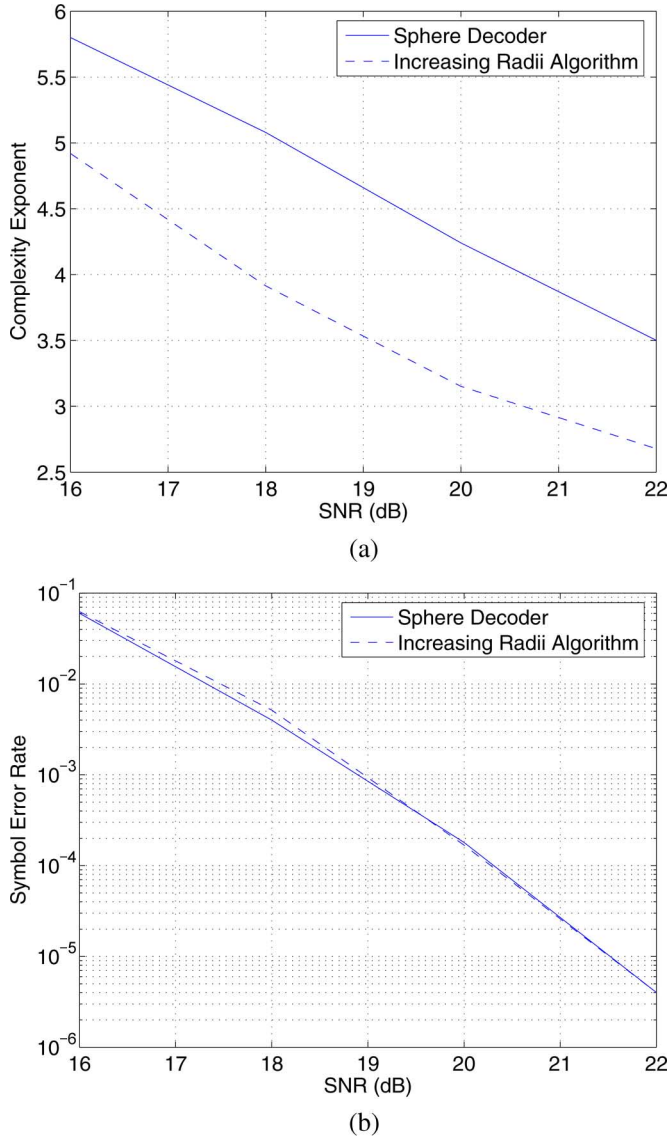


Fig. 5. Complexity Exponent and SER for the linear dispersion code with eight transmit and four receive antennas, with  $T = 8$ ,  $Q = 32$ , and  $R = 16$  with 16-QAM. From (a), we see that the IRA is 50 times faster than the sphere decoder on average. From (b), we see that the symbol error rates for the two algorithms are very close to each other, indicating no loss of performance. (a) Complexity exponent versus SNR; (b) SER versus SNR.

#### D. Simulations for the Upper Bound on the Complexity of IRA

We now compare the actual complexity of the increasing radii algorithm as obtained by simulations, with the upper bound derived in Theorem 1.

In Fig. 7, we present curves that show the complexity exponent for the increasing radii algorithm. For  $N$  being 20 and 60 and  $L = 2$  (4-QAM constellation) and SNR ranging from 5 to 30 dB we show the complexity exponent obtained through simulation, by using the upper bound of Theorem 1. We see that the upper bound is very good in this entire range. We also see that the simulated complexity can sometimes exceed the upper-bound. This is because the upperbound is on the expected complexity and need not hold for every instantiation of the decoding problem.

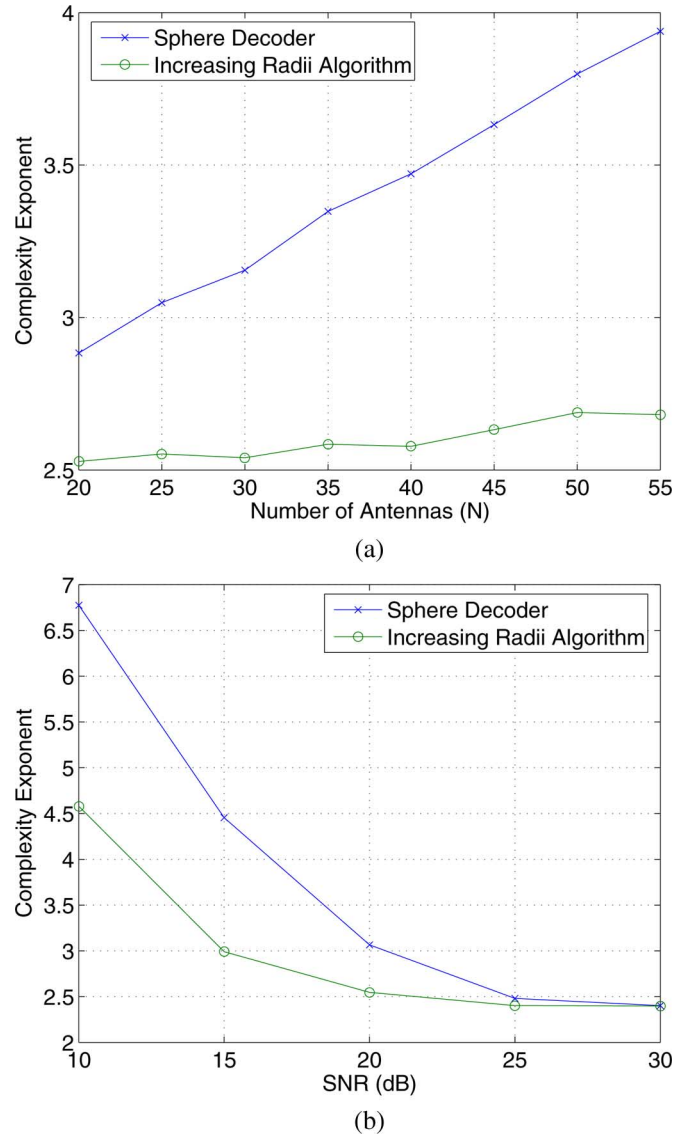


Fig. 6. Dependence of Complexity on  $N$  and SNR: (a) plots the complexities of the two algorithms against the number of antennas,  $N$ . The complexity exponent of the sphere decoder increases much faster than that of the IRA; (b) plots the two complexities against SNR. Computational savings with the IRA are more significant at low SNRs. (a) Dependence on  $N$ . SNR = 27 dB,  $L = 4$ . (b) Dependence on SNR.  $N = 50$ ,  $L = 2$ .

#### IX. CONCLUSION AND FUTURE WORK

In this paper, we have looked at the integer least-squares problem in a probabilistic setting. Because of this, the complexity of decoding is a random variable. Also, because of the statistics of the problem we are in a position to prune the search space so that we reduce the complexity while still keeping the transmitted point in the search region with high probability.

We have proposed a new method of doing this pruning and studied the complexity and the probability of error of the proposed method. The algorithm gives significant computational savings relative to the sphere decoder while still maintaining BERs close to optimal. For example, for a real problem in 100 dimensions, we can decode with up to 240 times less computation.



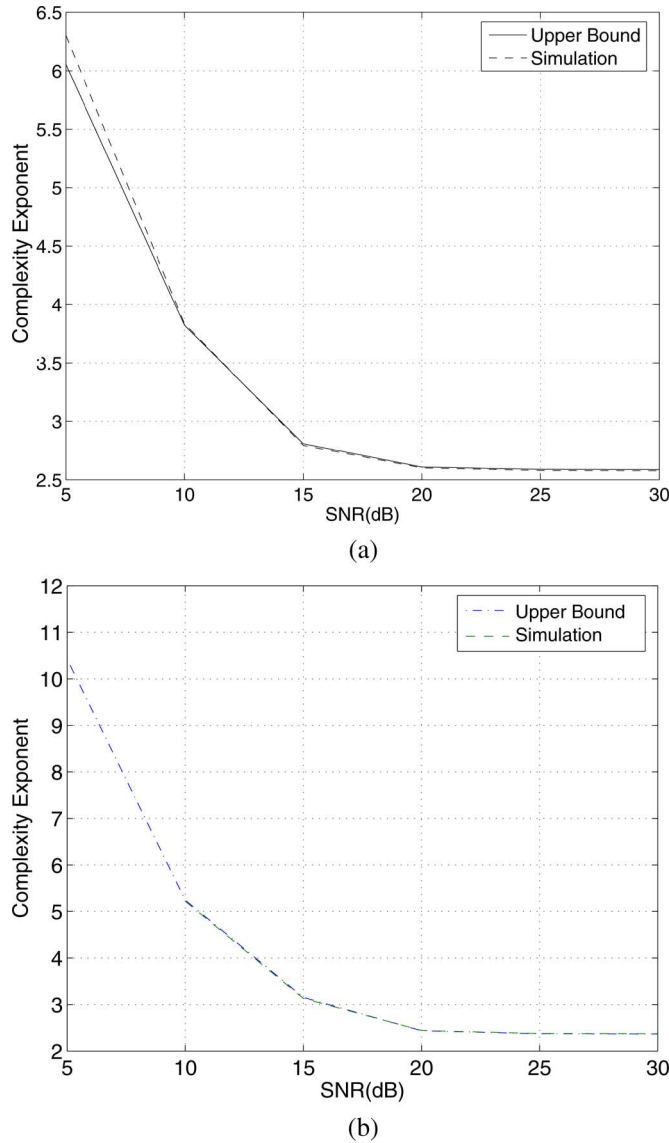


Fig. 7. Complexity Exponent for the IRA—simulated and upper bound. The simulations show that the complexity exponent for the IRA is tightly upper-bounded by Theorem 1. (a)  $N = 20, L = 2$ . Complexity exponent versus SNR. (b)  $N = 60, L = 2$ . Complexity exponent versus SNR.

Many interesting questions remain to be answered. Finding an optimal schedule for the IRA seems to be quite challenging since the complexity expressions we have are not exact, nor are they analytically tractable. Finding simpler expressions for the complexity as well as the BER would be of interest since these might help quantify more satisfactorily the tradeoff between performance and complexity and also give insight into optimizing the radii schedules.

The sphere decoding technique can be used for joint detection and decoding of block codes [28]. By analogy, the modified algorithms are also applicable in this context. Analysis of performance and complexity in this scenario is interesting. Another question of interest, which seems challenging, is the matter of choosing radii based on the known  $H$ . Clearly, the smallest region around  $x$  that contains the closest point depends on  $H$ , as well as  $v$ , but the current choice of  $r_i$  only takes the statistics of  $v$  into consideration.

We believe that the proposed pruning approach to the decoding problem demonstrates promise and that further work to analyze and optimize these statistical techniques will be of practical and theoretical interest.

## APPENDIX

### A. Appendix Derivation of Table (1)

For any  $u \in \mathbb{C}^{T \times 1}$ , we define  $u^i = [u_{T-i+1}, \dots, u_T]^T$ . Consider the  $H = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$  decomposition where  $H$  is  $N \times M$  with i.i.d.  $\mathcal{CN}(0, \sigma_h^2)$  entries,  $Q$  is unitary of size  $N \times N$  and  $R$  is upper triangular of size  $M \times M$ . It can be shown that the nondiagonal entries of  $R$  are i.i.d.  $\mathcal{CN}(0, \sigma_h^2)$  and the diagonal element  $R(i, i)$  is a scaled  $\chi$ -square distributed random variable (refer to [29]). More specifically,  $2R(i, i)/\sigma_h^2$  is  $\chi$ -square with  $2(N - i + 1)$  degrees of freedom. This means that it is the sum of squares of  $2(N - i + 1)$  i.i.d. standard real Gaussian random variables, i.e., variables having a  $\mathcal{N}(0, 1)$  distribution.

Therefore, a lower right submatrix of  $\begin{bmatrix} R \\ 0 \end{bmatrix}$  of size  $(i + N - M) \times i$ , say  $R_i$ , is statistically similar to it, i.e., it can be thought of as having arisen from the  $QR$  decomposition of an  $(i + N - M) \times i$  matrix  $H_i$  having i.i.d.  $\mathcal{CN}(0, \sigma_h^2)$  entries. Note that this is not to say that the  $H_i$  matrix is a submatrix of  $H$ . However, there exists  $H_i$  with the statistics mentioned above such that the  $QR$  decomposition of it gives us  $R_i$ , or,  $H_i = Q_i \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix}$  where  $Q_i$  is unitary of size  $(i + N - M) \times (i + N - M)$  (for more on this, refer to [13]).

Recall  $z$  from Section III. We have  $z = Q^*x - \begin{bmatrix} R \\ 0 \end{bmatrix}s = \begin{bmatrix} R \\ 0 \end{bmatrix}(\tilde{s} - s) + Q^*v$ . Define  $w = Q^*v$ . Clearly,  $w$  has the same statistics as  $v$ , i.e., i.i.d.  $\mathcal{CN}(0, 1)$  entries. Introduce  $v_i = Q_i w^{i+N-M}$ . Now  $v_i$  is of length  $(i + N - M)$  (it is not necessarily a subvector of  $v$ ). As in the case of  $w$ ,  $v_i$  will also have i.i.d.  $\mathcal{CN}(0, 1)$  entries. We can now write  $w^{i+N-M}$  as  $Q_i^*v_i$ .

Define  $\gamma_i = \sum_{j=1}^{i+N-M} \lambda_j$  for  $i = 1, \dots, M$ . Note that  $\gamma_i$  is the squared norm of  $z^{i+N-M}$ . Also, we have  $\tilde{s}^i$  and  $\tilde{s}^i$  as the lower length- $i$  subvectors of  $\tilde{s}$  and  $s$ , respectively. From the above arguments, we have  $z^{i+N-M} = \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix}(\tilde{s}^i - s^i) + Q_i^*v_i$ . Therefore

$$\begin{aligned} \gamma_i &= \|z^{i+N-M}\|^2 \\ &= \left\| \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix}(\tilde{s}^i - s^i) + Q_i^*v_i \right\|^2 \\ &= \left\| Q_i \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix}(\tilde{s}^i - s^i) + Q_i Q_i^*v_i \right\|^2 \\ &= \|H_i(\tilde{s}^i - s^i) + v_i\|^2 \end{aligned}$$

but it is clear that the vector  $H_i(\tilde{s}^i - s^i) + v_i$  has i.i.d.  $\mathcal{CN}(0, \sigma_v^2 + \sigma_h^2\|\tilde{s}^i - s^i\|^2)$ , i.e.,  $\mathcal{CN}(0, 1/c_i)$  entries. Therefore,  $\gamma_i$  is a scaled  $\chi$ -square distributed random variable. More specifically,  $2c_i\gamma_i$  is  $\chi$ -square with  $2i$  degrees of freedom. This means that it is the sum of squares of  $2i$  i.i.d. standard real Gaussian random variables, i.e., variables having a  $\mathcal{N}(0, 1)$  distribution. The expressions for the characteristic function of these are standard, and we have  $Ee^{j\alpha\gamma_i} = 1/(1 - (j\alpha/c_i)^{i+N-M})$ .

For  $\lambda_i$  where  $i > (N - M)$ , note that  $\gamma_{i-N+M} = \lambda_i + \gamma_{i-1-N+M}$ . Moreover, since the  $\lambda_i$ s are independent, so are  $\lambda_i$  and  $\gamma_{i-1-N+M}$ . Therefore,  $Ee^{j\alpha\gamma_{i-N+M}} = Ee^{j\alpha\lambda_i + j\alpha\gamma_{i-1-N+M}} = Ee^{j\alpha\lambda_i} Ee^{j\alpha\gamma_{i-1-N+M}}$ . Thus

$$Ee^{j\alpha\lambda_i} = \frac{Ee^{j\alpha\gamma_{i-N+M}}}{Ee^{j\alpha\gamma_{i-1-N+M}}} = \frac{\left(1 - \frac{j\alpha}{c_{i-1-N+M}}\right)^{i-1}}{\left(1 - \frac{j\alpha}{c_{i-N+M}}\right)^i}. \quad (\text{A1})$$

For  $i \leq (N - M)$  it is easy to see that  $\lambda_i$  is the squared norm of the  $(N - i + 1)$ th entry of  $Q^*v$ .  $Q^*v$  has the same statistics as  $v$ , i.e., i.i.d. entries, each with distribution  $\mathcal{CN}(0, 1)$ . With this, the characteristic function of  $\lambda_i$  is clearly  $1/(1 - (j\alpha/c_0))$ .

With this and the Fourier inversion, we get Table I.

### B. Derivation of Generating Function of Theorem 1

**Theorem 2:** For  $s, \tilde{s} \in \mathcal{S}^{k \times 1}$ , the number of solutions to  $\|s^k - \tilde{s}^k\|^2 = n$ , averaged over all possible values of  $\tilde{s}$  (as defined by  $r_k^L(n)$  in (22)) is given by the coefficient of  $x^n$  in  $(G_L(x))^k$  where  $G_L(x) = (1/L^2)(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2})^2$ . Recall that  $\mathcal{S} = \{a + jb | a, b \in \{-(L-1)/2, \dots, (L-3)/2, (L-1)/2\}\}$ .

*Proof:* For any complex vector  $x$  of length  $k$ , define the vector  $x_{\text{real}}$  as a real vector of length  $2k$  where  $x_{\text{real}}(2j-1) = \Re(x(j))$  and  $x_{\text{real}}(2j) = \Im(x(j))$  for  $j = 1, \dots, k$ .

Let  $r = s^k - \tilde{s}^k$  where  $s^k, \tilde{s}^k \in \mathcal{S}^{k \times 1}$ . Then define  $r_{\text{real}}$  as above. Also, define  $\mathcal{S}_{\text{real}} = \{-(L-1)/2, \dots, (L-3)/2, (L-1)/2\}$ .

Consider an arbitrary entry of  $r_{\text{real}}$ , say  $r_{\text{real}}(j)$ . For a fixed  $\tilde{s}^k$ ,  $\tilde{s}_{\text{real}}^k(j)$  is known. Say  $\tilde{s}_{\text{real}}^k(j) = t \in \mathcal{S}_{\text{real}}$ , then  $r_{\text{real}}(j)$  takes all values in  $\mathcal{S}_t = \mathcal{S}_{\text{real}} - t$ . Define  $q_t = \sum_{j \in \mathcal{S}_t} x^{j^2} \forall t \in \mathcal{S}_b$ . Associate with a fixed vector  $\tilde{s}^k$  the product  $q(\tilde{s}^k) = \prod_{j=1}^{2k} q_{\tilde{s}_{\text{real}}^k(j)}$ . Clearly, for this fixed  $\tilde{s}^k$ , the number of solutions to  $\|s^k - \tilde{s}^k\|^2 = n$  is the coefficient of  $x^n$  in  $q(\tilde{s}^k)$ .

Since all the  $L^{2k}$  possible  $\tilde{s}^k \in \mathcal{S}^{k \times 1}$  are assumed equally likely, the “average” number of solutions to  $\|s^k - \tilde{s}^k\|^2 = n$  is given by the coefficient of  $x^n$  in

$$\begin{aligned} & \frac{1}{L^{2k}} \sum_{\tilde{s}^k \in \mathcal{S}^{k \times 1}} q(\tilde{s}^k) \\ &= \frac{1}{L^{2k}} \sum_{\tilde{s}^k \in \mathcal{S}^{k \times 1}} \prod_{j=1}^{2k} q_{\tilde{s}_{\text{real}}^k(j)} \\ &= \frac{1}{L^{2k}} \sum_{t \in \mathcal{S}_{\text{real}}} \binom{2k}{\{\alpha_t | t \in \mathcal{S}_{\text{real}}\}} \prod_{t \in \mathcal{S}_{\text{real}}} q_t^{\alpha_t} \\ &= \frac{1}{L^{2k}} \left( \sum_{t \in \mathcal{S}_{\text{real}}} q_t \right)^{2i} \\ &= \frac{1}{L^{2k}} \left( L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2} \right)^{2i} \end{aligned}$$

where  $\binom{2k}{\{\alpha_t | t \in \mathcal{S}_{\text{real}}\}}$  is the multinomial coefficient given by  $\frac{(2k)!}{\alpha_{-(L-1)/2}! \alpha_{-(L-3)/2}! \dots \alpha_{(L-3)/2}! \alpha_{(L-1)/2}!}$ . Finally, we define  $G_L(x) = (1/L^2)(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2})^2$ .  $\square$

We note here that this is closely related to the problem of representing integers as a sum of squares. For more on this, refer to [13].

### REFERENCES

- [1] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.
- [2] B. Hassibi and B. Hochwald, “High-rate codes that are linear in space and time,” *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1804–1824, Jul. 2002.
- [3] M. O. Damen, A. Chkeif, and J.-C. Belfiore, “Lattice code decoder for space-time codes,” *IEEE Commun. Lett.*, vol. 4, no. 5, pp. 161–163, May 2000.
- [4] B. Hassibi, “An efficient square-root algorithm for BLAST,” in *Proc. IEEE Inf. Conf. Acoustics, Speech, Signal Process.*, Jun. 2000, vol. 2, pp. II737–II740.
- [5] R. Kannan, “Improved algorithms on integer programming and related lattice problems,” in *Proc. 15th Annu. ACM Symp. Theory of Computing*, 1983, pp. 193–206.
- [6] J. Lagarias, H. Lenstra, and C. Schnorr, “Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal,” *Combinatorica*, vol. 10, pp. 333–348, 1990.
- [7] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Math. Comput.*, vol. 44, pp. 463–471, April 1985.
- [8] J. Gross and J. Yellen, *Graph Theory Appl.*. Boca Raton, FL: CRC, 1998.
- [9] M. Stojnic, H. Vikalo, and B. Hassibi, “A branch and bound approach to speed up the sphere decoder,” in *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 2005, pp. iii/429–iii/432.
- [10] A. Banihashemi and A. Khandani, “On the complexity of decoding lattices using the Korkin-Zolotarev reduced basis,” *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 162–171, Mar. 1998.
- [11] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, “Closest point search in lattices,” *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
- [12] M. Ajtai, “Generating hard instances of lattice problems,” in *Proc. 28th Annual ACM Symp. Theory of Computing*, 1996, pp. 99–108.
- [13] H. Vikalo and B. Hassibi, “On the sphere decoding algorithm. I. Expected complexity II. Generalizations, second-order statistics, and applications to communications,” *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2834, Aug. 2005.
- [14] J. Jalden and B. Ottersten, “On the complexity of sphere decoding in digital communications,” *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.
- [15] R. Gowaikar and B. Hassibi, “Efficient statistical pruning for maximum likelihood decoding,” in *Proc. IEEE ICASSP*, 2003, pp. 49–52.
- [16] M. O. Damen, H. E. Gamal, and G. Caire, “On maximum likelihood detection and the search for the closest lattice point,” *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.
- [17] K. Su and I. J. Wassell, “A new ordering for efficient sphere decoding,” in *Proc. IEEE Int. Conf. Communications*, 2005, pp. 1906–1910.
- [18] C. Windpassinger, R. F. H. Fischer, and J. B. Huber, “Lattice-reduction-aided broadcast precoding,” *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2057–2060, Dec. 2004.
- [19] W. Xu, Y. Wang, Z. Zhou, and J. Wang, “Joint ML channel estimation and data detection for stbc via novel sphere decoding algorithms,” in *Proc. IEEE Vehicular Technology Conf.*, 2005, pp. 434–437.
- [20] W. Zhao and G. B. Giannakis, “Sphere decoding algorithms with improved radius search,” *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1104–1109, Jul. 2005.
- [21] R. Gowaikar and B. Hassibi, “Efficient near-ml decoding via statistical pruning,” in *Proc. IEEE Int. Symp. Information Theory*, 2003, p. 274.
- [22] B. Hochwald and S. ten Brink, “Achieving near-capacity on a multiple-antenna channel,” *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
- [23] M. Grotschel, L. Lovasz, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed. New York: Springer-Verlag, 1993.
- [24] J. Y. Cai, “On the average-case hardness of CVP,” in *Proc. IEEE Symp. Foundations of Computer Science*, 2001, pp. 308–317.
- [25] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Boelcskei, “Vlsi implementation of mimo detection using the sphere decoding algorithm,” *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.

- [26] R. Gowaikar and B. Hassibi, "Statistical pruning for near-maximum likelihood decoding," Tech. Rep. [Online]. Available: <http://www.ee.caltech.edu/gowaikar/pubs/report.pdf>
- [27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [28] H. Vikalo and B. Hassibi, "On joint ML detection and decoding for linear block codes," in *Proc. IEEE ISIT*, 2003, p. 275.
- [29] A. Edelman, "Eigenvalues and condition numbers of random matrices," Ph.D. thesis, Dept. Math., Massachusetts Inst. Technol., Cambridge, 1989.



**Radhika Gowaikar** (S'03) received the B.Tech. degree from the Indian Institute of Technology, Bombay, in 2001, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, in 2002 and 2006, respectively, all in electrical engineering.

Her research interests include sensor and *ad hoc* networks, network coding for wireless networks, and decoding in multiple antenna systems.



**Babak Hassibi** was born in Tehran, Iran, in 1967. He received the B.S. degree from the University of Tehran in 1989 and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1993 and 1996, respectively, all in electrical engineering.

From October 1996 to October 1998, he was a Research Associate at the Information Systems Laboratory, Stanford University, and from November 1998 to December 2000, he was a Member of the Technical Staff in the Mathematical Sciences Research Center at Bell Laboratories, Murray Hill, NJ. Since January

2001, he has been with the Department of Electrical Engineering, California Institute of Technology, Pasadena, where he is currently an Associate Professor. He has also held short-term appointments at the Ricoh California Research Center, the Indian Institute of Science, and Linköping University, Sweden. He is the coauthor of the books *Indefinite Quadratic Estimation and Control: A Unified Approach to  $H^2$  and  $H^\infty$  Theories* (SIAM, 1999) and *Linear Estimation* (Prentice-Hall, 2000). His research interests include wireless communications, robust estimation and control, adaptive signal processing, and linear algebra.

Dr. Hassibi is a recipient of an Alborz Foundation Fellowship, the 1999 O. Hugo Schuck best paper award of the American Automatic Control Council, the 2002 National Science Foundation Career Award, the 2002 Okawa Foundation Research Grant for Information and Telecommunications, the 2003 David and Lucille Packard Fellowship for Science and Engineering, and the 2003 Presidential Early Career Award for Scientists and Engineers (PECASE). He was a Guest Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY Special Issue on Space-Time Transmission, Reception, Coding, and Signal Processing, and an Associate Editor of Communications for the IEEE TRANSACTIONS ON INFORMATION THEORY for 2003 to 2006.